Supplementary Material
Prompting Visual-Language Models for Efficient Video Understanding

Chen Ju[1], Tengda Han[2], Kunhao Zheng[1], Ya Zhang[1] ✉, and Weidi Xie[1,2] ✉

[1] Cooperative Medianet Innovation Center, Shanghai Jiao Tong University
[2] Visual Geometry Group, University of Oxford
`{ju_chen, dyekuu, ya_zhang, weidi}@sjtu.edu.cn` `htd@robots.ox.ac.uk`

1 Implementation Details

The image and text encoders are adopted from the pre-trained CLIP (ViT-B/16), and are both kept frozen. All prompt vectors and visual features are of the same dimension, $D = 512$, and the temperature hyper-parameter τ is set to 0.07. Both prompt vectors and temporal Transformer are randomly initialized by drawing from a zero-mean Gaussian distribution with the standard deviation of 0.01. For action recognition and action localisation (the second-stage proposal classifier), we evaluate different numbers of prompt vectors, and adopt the $[16+16]$ pattern eventually, *i.e.* 16 random vectors are prepended / appended to the input text tokens, and optimised for the considered tasks. For text-video retrieval, as the text description can be long, we utilise $[4+4]$ prompt vectors. In terms of spatial pre-processing, we resize the frame's short side to 224, while keeping its original aspect ratio, then perform center cropping to convert the spatial size to 224×224. The maximum number of textual tokens is 77 (follow the official CLIP design), and the temperature hyper-parameter τ is set to 0.07.

1.1 Action Recognition

For action recognition, all videos are decoded to 30 fps, and each video is sampled 16 frames with a random frame gap (gap $\in \{1, 2, 3, 4, 5, 6, 10, 15\}$) for training [19]. The temporal positional encodings consist of each frame's index and the frame sampling gap (video playing speed). The model is optimised using AdamW [14] with a learning rate of 10^{-4}, and a batch size of 64 videos. During inference, we random sample 16 frames from each video for 5 times, and take the average of these five results as the final predictions, *i.e.* 5-crop evaluation.

1.2 Action Localisation

For action localisation, to obtain class-agnostic action proposals, we adopt the off-the-shelf proposal detectors [10, 20].

To be specific, we first divide the entire video into several equal-frame segments; use the CLIP image encoder with one Transformer layer to extract frame-wise embeddings; feed these embeddings to the 6-layer feature pyramid; utilise

Table 1: **Results of proposal detection.** For the closed-set scenario, we train and evaluate on the same action categories. While for the zero-shot scenario, we experiment on two settings: training with 75% (25%) categories and testing on the remaining 25% (50%) action categories.

		THUMOS14		ActivityNet1.3
Scenario	Train *v.s* Test	AR@50	AR@100	AR@100
Closed-set	100% *v.s* 100%	32.4	38.3	63.6
Zero-shot	75% *v.s* 25%	24.1	29.7	60.8
Zero-shot	50% *v.s* 50%	21.2	26.2	59.3

three parallel prediction heads to determine the actionness, centerness, boundaries respectively; finally, assemble all prediction results and use Soft-NMS [3] to suppress redundant proposals. On ActivityNet1.3, we maintain the original video frame rate, and use 768 frames in each segment. On THUMOS14, we downsample each video to 10 fps, and 256 frames are used to construct the segment. The proposal detector is optimised using AdamW [14] with a learning rate of 10^{-4}, and a batch size of 32 videos. Please refer to [10, 20] for detailed architectures and optimisation objectives. For post-processing, we set the tIoU threshold in Soft-NMS to 0.5 on THUMOS14, and 0.85 on ActivityNet1.3.

1.3 Text-Video Retrieval

For text-video retrieval, all the videos are decoded with 30 fps in advance, and we take the 16-frame input with a random frame gap $\in \{10, 15, 30\}$, that is, the video is equivalent to being sampled with 1-3 fps. Note that, here we adopt significantly lower fps than action recognition, as the video retrieval task tends to require information from long-term visual dependencies.

2 Experimental Results

In this section, we demonstrate more results to further analyse our method, and explore the semantic information learnt by prompt vectors.

2.1 Action Localisation

We adopt the two-stage paradigm for action localisation, *i.e.* first-stage proposal detection and second-stage proposal classification. In this section, we separately evaluate the performance of these two stages in closed-set and zero-shot scenarios, to comprehensively dissect localisation results.

Table 2: **Results of proposal classification.** For closed-set scenarios of THUMOS14 (ActivityNet1.3), we train and evaluate on the same 20 (200) categories. While for zero-shot, we experiment with two settings, training with 75% (50%) action categories and testing on the left 25% (50%) action categories, *e.g.* training on 15 (10) categories and testing on the left 5 (10) categories for THUMOS14.

		THUMOS14		ActivityNet1.3	
Scenario	Train *v.s* Test	train / test	TOP1	train / test	TOP1
Closed-set	100% *v.s* 100%	20 / 20	88.7	200 / 200	85.6
Zero-shot	75% *v.s* 25%	15 / 5	93.4	150 / 50	81.5
Zero-shot	50% *v.s* 50%	10 / 10	87.3	100 / 100	71.8

Proposal Detection. To evaluate the class-agnostic action proposals, we adopt conventional metric: Average Recall with different Average Number (AR@AN). On THUMOS14, the AR is calculated under multiple IoU threshold set from 0.5 to 1.0 with a stride of 0.05. As for ActivityNet1.3, the multiple IoU threshold are from 0.5 to 0.95 with a stride of 0.05. And for the zero-shot settings with multiple sampling trials, we average the AR of all trials.

Table 1 shows the comparison results. On both datasets, the performance of the zero-shot scenario decreases compared with that of the closed-set scenario, showing that the action proposal is in fact not perfectly class-agnostic, it is still biased towards seen action categories. Moreover, since each video on THUMOS14 contains denser action instances, the number of which is 10 times than that of ActivityNet1.3, the performance drop on THUMOS14 is more significant.

Proposal Classification. We here eliminate the action proposals that are completely disjoint with all ground-truth action instances, and evaluate the standard TOP1 classification accuracy among the remaining action proposals.

Table 2 shows the average accuracy of multiple sampling trials. Comparing to the closed-set evaluation, the zero-shot classification accuracy tends to drop. Note that, the setting training with 75% action categories on THUMOS14 is a special case. Since THUMOS14 has total 20 action categories, in this case, the number of testing categories is only 5, thus the classification task is definitely easier than the closed-set scenario.

Summary. The above results show that, the performance drop of the zero-shot scenario comes from two sources: one is the recall drop from the first-stage action proposals, and the other comes from the second-stage classification errors.

2.2 Text-Video Retrieval

Here, we add more comparison results for retrieval benchmarks. Since the text encoder from the pre-trained CLIP takes limited number of textual tokens up to

Table 3: **Results of text-video retrieval.** E2E denotes if the model is trained end-to-end. Baseline-IV refers to the original CLIP with text query naïvely encoded, *i.e.* without adopting any prompt. We highlight the results without end-to-end finetuning, where the best and second-best results are highlighted with bold and underline. As these methods are pre-trained on different datasets with variable sizes, it is unlikely to make fair comparisons.

		MSRVTT(9K)				LSMDC				DiDeMo	
Method	E2E	R@1↑	R@5↑	R@10↑	MdR↓	R@1↑	R@5↑	R@10↑	MdR↓	R@1↑	R@5↑
CE [12]	✗	21.7	51.8	65.7	5.0	12.4	28.5	37.9	21.7	16.1	41.1
MMT [7]	✗	24.6	54.0	67.1	4.0	13.2	29.2	38.8	21.0	–	–
TT-CE+ [4]	✗	29.6	61.6	74.2	3.0	17.2	36.5	46.3	13.7	21.6	48.6
SMiT [17]	✗	33.1	64.8	77.4	–	–	–	–	–	–	–
MDMMT [5]	✗	**38.9**	**69.0**	**79.7**	**2.0**	**18.8**	**38.5**	**47.9**	**12.3**	–	–
Baseline-IV	✗	31.2	53.7	64.2	4.0	11.3	22.7	29.2	56.5	28.8	54.6
Ours	✗	36.7	64.6	76.8	**2.0**	13.4	29.5	40.3	18.6	**36.1**	**64.8**
Frozen [2]	✓	31.0	59.5	70.5	3.0	15.0	30.8	39.8	20.0	34.6	65.0
CLIP4Clip [15]	✓	44.5	71.4	81.6	2.0	22.6	41.0	49.1	11.0	43.4	70.2

77, whereas the text query of retrieval can be long, in these experiments, we only employ 8 learnable prompt vectors, *i.e.* [4+X+4]. And for temporal modeling, we only use two Transformer layers to achieve efficient model adaptation.

As can be observed in Table 3, when comparing to the Baseline-IV, which denotes the original CLIP using naïvely-encoded text queries, our proposed prompt learning and temporal modeling clearly demonstrate benefits on all benchmarks. Additionally, while comparing to the existing approaches that are specifically targeting on retrieval, our method also performs competitively.

Note that, we try to compare with the results reported in the existing work, however, this is by no means to be fair comparisons, as these methods are usually pre-trained on different datasets with variable sizes. For instance, CLIP4Clip [15] is pre-trained on HowTo100M [16] (136M videos). Thus, our method is naturally at a disadvantage by only training on small-scale datasets. Moreover, in terms of computation cost, our method only optimises several prompt vectors, along with two Transformer layers, and all experiments can be done on *one* 24G GeForce RTX 3090 GPU. While CLIP4Clip needs to end-to-end finetune the CLIP backbone, with *four* 32G Tesla V100 GPUs, costing significantly more.

2.3 Fine-grained Action Recognition

Here, we further evaluate our method on the popular fine-grained motion dataset: Something-Something V2 [8]. It contains $220,847$ videos with 174 action categories. The standard split is $168,913$ training videos, $24,777$ validation videos, and $27,157$ testing videos. Some categories are very fine-grained, *e.g.* bending something so that it deforms *v.s* bending something until it breaks.

Table 4: **Something-Something V2 closed-set results**. Baseline-V refers to the "zero-shot" CLIP inference without using any prompt template. TFM refers to the number of Transformer layers for temporal modeling.

Method	Prompt	Temporal	TOP1
TRN [22]	✗	✗	48.8
SlowFast [6]	✗	✗	61.7
TSM [11]	✗	✗	63.4
ViVIT [1]	✗	✗	65.9
Swin-B [13]	✗	✗	69.6
Baseline-V	✗	✗	1.4
C1	16+X+16	✗	18.4
C2	16+X+16	4-TFM	38.1

Table 4 reports the ablation and comparison results. Comparing to the simple Baseline-V, *i.e.* the "zero-shot" CLIP inference without any prompt, both prompt learning and temporal Transformer bring considerable performance gains. However, there is still a certain gap between our results and existing state-of-the-art methods, we conjecture that this may be due to that, the CLIP pre-training relies more on object information for action recognition, lacking prior knowledge of fine-grained motions.

2.4 Prompt Semantics

To demonstrate the prompt semantics, we visualise the learnt 32 prompt vectors, by searching for the word embeddings whose cosine distance is nearest to them. Here, we regard the CLIP vocabulary library as the total search set, *i.e.* 49408 subwords. For HMDB51 under the closed-set scenario, the nearest subwords are "*educ, Ā, giggle, meyers, lucas, windows, resolution, fives, me, lump, chancellor, extensively, previously, trades, sden, bowler, giuliani, radi, ivory, ffey, plays, evolu, acies, ghead, forsyth, botanic, unite, &, protestant, saucer, ferry, mango*".

As can be seen, some searched subwords are related to datasets or tasks, but most do not correspond to meaningful semantics. Such phenomenon about learnt prompt semantics, is in accordance with the observation in the NLP domain [9]. We speculate this is because, the prompt vectors learnt in continuous embedding space go beyond discrete vocabulary space. In other words, the CLIP vocabulary library is limited to interpret the learnt prompt semantics.

3 Limitations

Our proposed idea relies on the visual-language model pre-trained on the large-scale image alt-text data, which could potentially incur two limitations: *First*, bias in the web data. *Second*, as temporal modeling is only used on top of visual features, it may fail to model fine-grained motions.

4 Dataset Splits

Here, we detail the dataset splits for training and testing, under different scenarios, namely, few-shot action recognition, zero-shot action recognition, and zero-shot action localisation. All the splits can be available at https://github.com/ju-chen/Efficient-Prompt/tree/main/datasplits.

4.1 Few-shot Action Recognition

5-Shot-5-Way Setting. We here adopt the publicly available few-shot data splits, *i.e.* sample 5 action categories (5 videos per category) from a set of testing categories, to form the few-shot support set. We conduct 200 trials with random samplings, to ensure the statistical significance.

- **Kinetics-400**. We follow [23, 18] and sample the test action categories from: blasting sand, busking, cutting watermelon, dancing ballet, dancing charleston, dancing macarena, diving cliff, filling eyebrows, folding paper, hula hooping, hurling (sport), ice skating, paragliding, playing drums, playing monopoly, playing trumpet, pushing car, riding elephant, shearing sheep, side kick, stretching arm, tap dancing, throwing axe, unboxing.
- **UCF-101**. Following [21], the test action categories are sampled from : blowingcandles, cleanandjerk, cliffdiving, cuttinginkitchen, diving, floorgymnastics, golfswing, handstandwalking, horserace, icedancing, jumprope, pommelhorse, punch, rockclimbingindoor, salsaspin, skiing, skydiving, stillrings, surfing, tennisswing, volleyballspiking.
- **HMDB-51**. We follow [21] and sample the test action categories from: fencing, kick, kick ball, pick, pour, pushup, run, sit, smoke, talk.

5-Shot-C-Way Setting. In this generalised problem, to construct the dataset for training, we sample 5 videos from all categories and measure the performance on the standard testing set, *i.e.* all videos from all categories in the testing set. We also conduct 10 random sampling rounds to choose training videos.

- **Kinetics-400**. Its training set contains 2000 videos, *i.e.* 400×5 videos, and the testing set covers 19101 videos.
- **UCF-101**. The training set contains 505 videos, *i.e.* 101×5 videos, and the testing set covers 3783 videos.
- **HMDB-51**. The training data covers 255 videos, *i.e.* 51×5 videos, and the testing set contains 1530 videos.

4.2 Zero-shot Action Recognition

In this section, we split K-700 dataset into two subsets with disjoint categories. Specifically, 400 action categories are used for training, and the remaining 300 action categories are used for evaluation.

- **Training Categories (#400):** carving wood with a knife, cracking neck, feeding goats, fixing bicycle, passing soccer ball, being in zero gravity, breaking boards, changing gear in car, playing organ, taking photo, finger snapping, walking on stilts, cleaning shoes, hoverboarding, putting wallpaper on wall, using atm, rock scissors paper, riding elephant, running on treadmill, cracking back, pulling rope (game), washing feet, skydiving, country line dancing, throwing knife, square dancing, fixing hair, folding clothes, doing jigsaw puzzle, making slime, using a power drill, welding, jumping jacks, cosplaying, surveying, bottling, smoking pipe, shooting basketball, swimming with dolphins, tying bow tie, cleaning gutters, playing cards, playing dominoes, uncorking champagne, drop kicking, folding paper, standing on hands, massaging neck, swing dancing, chopping meat, breading or breadcrumbing, laying concrete, driving car, sawing wood, clean and jerk, embroidering, pinching, playing saxophone, tango dancing, peeling banana, drumming fingers, throwing axe, lawn mower racing, roller skating, celebrating, dyeing eyebrows, arm wrestling, belly dancing, using segway, playing cello, news anchoring, mountain climber (exercise), treating wood, riding mechanical bull, cutting watermelon, playing laser tag, picking apples, using a sledge hammer, skipping rope, feeding fish, playing basketball, carving pumpkin, bee keeping, holding snake, walking through snow, fly tying, tightrope walking, playing monopoly, shopping, planing wood, brushing floor, cleaning pool, spinning poi, grooming horse, laughing, sign language interpreting, roasting pig, making cheese, ripping paper, decorating the christmas tree, spraying, snowkiting, putting on shoes, playing cricket, ironing, mosh pit dancing, swimming butterfly stroke, ironing hair, making the bed, chiseling stone, javelin throw, playing keyboard, poaching eggs, playing recorder, blowing nose, high kick, shot put, tasting beer, laying tiles, making paper aeroplanes, being excited, parkour, playing piano, throwing discus, wading through mud, washing dishes, headbutting, tying knot (not on a tie), unloading truck, visiting the zoo, picking blueberries, gymnastics tumbling, playing checkers, hugging baby, playing netball, spray painting, attending conference, playing trombone, using bagging machine, listening with headphones, making sushi, trimming or shaving beard, swimming with sharks, throwing water balloon, plastering, playing pan pipes, directing traffic, assembling computer, making horseshoes, ice swimming, pull ups, battle rope training, blowdrying hair, doing laundry, ice skating, shouting, surfing water, barbequing, vacuuming floor, squat, dribbling basketball, chasing, throwing ball (not baseball or American football), eating doughnuts, contact juggling, deadlifting, dancing gangnam style, pretending to be a statue, shaving head, putting on eyeliner, blowing bubble gum, jumping into pool, juggling fire, grinding meat, moving furniture, tagging graffiti, skiing mono, bookbinding, walking the dog, petting animal (not cat), falling off bike, scrambling eggs, sipping cup, separating eggs, historical reenactment, springboard diving, eating watermelon, card throwing, using a microscope, playing poker, making pizza, assembling bicycle, backflip (human), seasoning food, getting a tattoo, shining shoes, snatch weight lifting, installing carpet, getting a haircut, laying decking, rock climbing, sieving, rope pushdown, opening bottle (not wine), salsa dancing, catching or throwing baseball, texting, clapping, mopping floor, pirouetting, scuba diving, coughing, climbing a rope, changing oil, yarn spinning, playing guitar, using a paint roller, snowmobiling, tying necktie, vacuuming car, petting horse, busking, paragliding, playing kickball, chewing gum, giving or receiving award, drooling, putting in contact lenses, alligator wrestling, doing aerobics, whistling, somersaulting, carrying baby, decoupage, slicing onion, jetskiing, carving ice, baking cookies, checking watch, rolling pastry, pumping fist, crocheting, eating burger, jumping sofa, dodgeball, karaoke, waxing back, leatherworking, passing American football (not in game), massaging feet, dumpster diving, making balloon shapes, cracking knuckles, eating spaghetti, catching or throwing frisbee, drinking shots, playing gong, acting in play, shoveling snow, sharpening knives, using megaphone, doing nails, burping, inflating balloons, flying kite, herding cattle, doing sudoku, eating hotdog, putting on sari, punching bag, singing, squeezing orange, pushing cart, splashing water, playing trumpet, exercising arm, fencing (sport), ski jumping, lock picking, carrying weight, using inhaler, waking up, staring, photobombing, eating carrots, bungee jumping, checking tires, weaving fabric, home roasting coffee, playing didgeridoo, getting a piercing, building cabinet, jumping bicycle, capoeira, reading newspaper, playing rubiks cube, high jump, raising eyebrows, stretching arm, shooting off fireworks, dancing charleston, pillow fight, hockey stop, steering car, drawing, recording music, front raises, riding camel, wrapping present, waxing legs, sleeping, cooking scallops, sucking lolly, cutting cake, threading needle, base jumping, dining, trapezing, tackling, building shed, tiptoeing, cooking chicken, playing harmonica, training dog, setting table, curling eyelashes, passing American football (in game), docking boat, playing paintball, sneezing, playing with trains, swimming breast stroke, sticking tongue out, cutting pineapple, lunge, triple jump, marriage proposal, cleaning windows, diving cliff, bench pressing, making a cake, saluting, luge, driving tractor, swimming front crawl, bending back, laying stone, pushing car, sanding wood, dunking basketball, sanding floor, sausage making, robot dancing, building sandcastle, tasting food, spelunking, baby waking up, playing darts, playing american football, land sailing, sword fighting, ski ballet, playing mahjong, smelling feet, blasting sand, peeling potatoes, smoking, hurdling, grooming cat, pouring beer, bobsledding, flint knapping, washing hands, clay pottery making, digging, air drumming, moving child, fidgeting, packing, delivering mail, skipping stone, cartwheeling, playing bass guitar, tai chi, using remote controller (not gaming), playing pinball, bartending, waxing chest, parasailing, egg hunting, carving marble, wrestling, snowboarding, headbanging, playing hand clapping games, abseiling, crawling baby, skiing slalom, frying vegetables, wading through water.

- **Testing Categories (#300):** adjusting glasses, answering questions, applauding, applying cream, archaeological excavation, archery, arguing, arranging flowers, arresting, auctioning, bandaging, bathing dog, beatboxing, bending metal, biking through snow, blending fruit, blowing glass, blowing leaves, blowing out candles, bodysurfing, bouncing ball (not juggling), bouncing on bouncy castle, bouncing on trampoline, bowling, braiding hair, breakdancing, breaking glass, breathing fire, brushing hair, brushing teeth, brush painting, building lego, bulldozing, calculating, calligraphy, canoeing or kayaking, capsizing, card stacking, casting fishing line, catching fish, catching or throwing softball, changing wheel (not on bike), cheerleading, chiseling wood, chopping wood, clam digging, cleaning toilet, climbing ladder, climbing tree, closing door, coloring in, combing hair, contorting, cooking egg, cooking on campfire, cooking sausages (not on barbeque), counting money, crossing eyes, crossing river, crying, cumbia, curling hair, curling (sport), cutting apple, cutting nails, cutting orange, dancing ballet, dancing macarena, dealing cards, disc golfing, dyeing hair, eating cake, eating chips, eating ice cream, eating nachos, entering church, exercising with an exercise ball, extinguishing fire, faceplanting, falling off chair, feeding birds, filling cake, filling eyebrows, flipping bottle, flipping pancake, folding napkins, gargling, geocaching, gold panning, golf chipping, golf driving, golf putting, gospel singing in church, grooming dog, hammer throw, hand washing clothes, head stand, helmet diving, high fiving, hitting baseball, hopscotch, huddling, hugging (not baby), hula hooping, hurling (sport), ice climbing, ice fishing, jaywalking, jogging, juggling balls, juggling soccer ball, jumpstyle dancing, kicking field goal, kicking soccer ball, kissing, kitesurfing, knitting, krumping, laying bricks, letting go of balloon, licking, lifting hat, lighting candle, lighting fire, longboarding, long jump, looking at phone, looking in mirror, making a sandwich, making bubbles, making jewelry, making latte art, making snowman, making tea, marching, massaging back, massaging legs, massaging person's head, metal detecting, milking cow, milking goat, mixing colours, moon walking, motorcycling, moving baby, mowing lawn, mushroom foraging, needle felting, opening coconuts, opening door, opening present, opening refrigerator, opening wine bottle, peeling apples, person collecting garbage, petting cat, photocopying, planting trees, playing accordion, playing badminton, playing bagpipes, playing beer pong, playing billiards, playing blackjack, playing chess, playing clarinet, playing controller, playing cymbals, playing drums, playing field hockey, playing flute, playing harp, playing ice hockey, playing lute, playing maracas, playing marbles, playing nose flute, playing oboe, playing ocarina, playing piccolo, playing ping pong, playing polo, playing road hockey, playing rounders, playing scrabble, playing shuffleboard, playing slot machine, playing squash or racquetball, playing tennis, playing ukulele, playing violin, playing volleyball, playing xylophone, poking bellybutton, pole vault, polishing furniture, polishing metal, popping balloons, pouring milk, pouring wine, preparing salad, presenting weather forecast, pulling espresso shot, pumping gas, punching person (boxing), pushing wheelbarrow, pushing wheelchair, push up, putting on foundation, putting on lipstick, putting on mascara, reading book, repairing puncture, riding a bike, riding mule, riding or walking with horse, riding scooter, riding snow blower, riding unicycle, roasting marshmallows, rolling eyes, sailing, scrapbooking, scrubbing face, sewing, shaking hands, shaking head, shaping bread dough, sharpening pencil, shaving legs, shearing sheep, shining flashlight, shoot dance, shooting goal (soccer), shredding paper, shucking oysters, shuffling cards, shuffling feet, side kick, silent disco, situp, skateboarding, skiing crosscountry, slacklining, slapping, sled dog racing, smashing, smoking hookah, snorkeling, spinning plates, stacking cups, stacking dice, steer roping, stomping grapes, stretching leg, surfing crowd, sweeping floor, swimming backstroke, swinging baseball bat, swinging on something, sword swallowing, talking on cell phone, tap dancing, tapping guitar, tapping pen, tasting wine, testifying, throwing snowballs, throwing tantrum, tickling, tie dying, tobogganing, tossing coin, tossing salad, trimming shrubs, trimming trees, twiddling fingers, tying shoe laces, unboxing, using a wrench, using circular saw, using puppets, waiting in line, walking with crutches, washing hair, watching tv, watering plants, water skiing, water sliding, waving hand, waxing armpits, waxing eyebrows, weaving basket, windsurfing, winking, wood burning (art), writing, yawning, yoga, zumba.

4.3 Zero-shot Action Localisation

Here, we initiate two evaluation settings on THUMOS14 and ActivityNet1.3: (A) train on 75% action categories and test on the remaining 25% action categories; (B) train on 50% categories and test on the remaining 50% categories.

For setting (A) on THUMOS14, the number of training and testing categories is 15 and 5, respectively. For setting (B) on THUMOS14, the number of both training and testing action categories is 10. For setting (A) on ActivityNet1.3, the number of training and testing categories is 150 and 50. For setting (B) on ActivityNet1.3, the number of both training and testing categories is 100.

Under each setting, we conduct 10 random samplings to split categories for training and testing. Note that, as untrimmed videos in localisation are normally minutes long, splitting datasets based on action categories may incur some situations, where the same video contains both training and testing categories. For this multi-label video, we simply divide it into two videos, one containing only training categories and the other containing only testing categories.

References

1. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: Vivit: A video vision transformer. In: Proceedings of the International Conference on Computer Vision (2021)
2. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: Frozen in time: A joint video and image encoder for end-to-end retrieval. Proceedings of the International Conference on Computer Vision (2021)
3. Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-nms–improving object detection with one line of code. In: Proceedings of the International Conference on Computer Vision (2017)
4. Croitoru, I., Bogolin, S.V., Leordeanu, M., Jin, H., Zisserman, A., Albanie, S., Liu, Y.: Teachtext: Crossmodal generalized distillation for text-video retrieval. In: Proceedings of the International Conference on Computer Vision (2021)
5. Dzabraev, M., Kalashnikov, M., Komkov, S., Petiushko, A.: Mdmmt: Multidomain multimodal transformer for video retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (2021)
6. Feichtenhofer, C.: X3d: Expanding architectures for efficient video recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
7. Gabeur, V., Sun, C., Alahari, K., Schmid, C.: Multi-modal transformer for video retrieval. In: Proceedings of the European Conference on Computer Vision (2020)
8. Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al.: The "something something" video database for learning and evaluating visual common sense. In: Proceedings of the International Conference on Computer Vision (2017)
9. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. In: Proceedings of the Conference on Empirical Methods in Natural Language Processinng (2021)
10. Lin, C., Xu, C., Luo, D., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Fu, Y.: Learning salient boundary feature for anchor-free temporal action localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2021)
11. Lin, J., Gan, C., Han, S.: Tsm: Temporal shift module for efficient video understanding. In: Proceedings of the International Conference on Computer Vision (2019)
12. Liu, Y., Albanie, S., Nagrani, A., Zisserman, A.: Use what you have: Video retrieval using representations from collaborative experts. In: Proceedings of the British Machine Vision Conference (2019)
13. Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H.: Video swin transformer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2022)
14. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: Proceedings of the International Conference on Learning Representations (2019)
15. Luo, H., Ji, L., Zhong, M., Chen, Y., Lei, W., Duan, N., Li, T.: CLIP4Clip: An empirical study of clip for end to end video clip retrieval. arXiv preprint arXiv:2104.08860 (2021)
16. Miech, A., Zhukov, D., Alayrac, J.B., Tapaswi, M., Laptev, I., Sivic, J.: Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In: Proceedings of the International Conference on Computer Vision (2019)

17. Monfort, M., Jin, S., Liu, A., Harwath, D., Feris, R., Glass, J., Oliva, A.: Spoken moments: Learning joint audio-visual representations from video descriptions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2021)
18. Perrett, T., Masullo, A., Burghardt, T., Mirmehdi, M., Damen, D.: Temporal relational crosstransformers for few-shot action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2021)
19. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Val Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: Proceedings of the European Conference on Computer Vision (2016)
20. Yang, L., Peng, H., Zhang, D., Fu, J., Han, J.: Revisiting anchor mechanisms for temporal action localization. IEEE Transactions on Image Processing (2020)
21. Zhang, H., Zhang, L., Qi, X., Li, H., Torr, P.H.S., Koniusz, P.: Few-shot action recognition with permutation-invariant attention. In: Proceedings of the European Conference on Computer Vision (2020)
22. Zhou, B., Andonian, A., Oliva, A., Torralba, A.: Temporal relational reasoning in videos. In: Proceedings of the European Conference on Computer Vision (2018)
23. Zhu, L., Yang, Y.: Compound memory networks for few-shot video classification. In: Proceedings of the European Conference on Computer Vision (2018)